NEXT-GENERATION CONVERSATIONAL AI FOR DATABASE ACCESS: THE TECHNOLOGY BEHIND AUTOQL

HOW INNOVATION IN CONVERSATIONAL AI IS REVOLUTIONIZING THE WAY TODAY'S LEADING BUSINESSES ACCESS AND LEVERAGE THEIR DATA.

CHATA

Table of Contents

Overview

This white paper is for software developers and solution providers looking to understand the value of the technology we've developed to build AutoQL into a powerful conversational AI solution for database access. In this white paper, you'll learn about:

- The current limitations of intent classification systems as they pertain to the database access use case.

- What it takes to go beyond current technologies and develop conversational AI that can cater to the data access and seamless experiences today's users demand.

- How implementing a conversational AI solution for database access adds value to any software application or business portal that creates and collects data.

Introduction

In today's competitive market, data has emerged as a key differentiating factor, separating the businesses that stay agile and experience success, from those that struggle to find footing in a volatile economy. The ability to leverage data to make better decisions is increasingly top-of-mind for industry leaders looking to get ahead and stay ahead.

Thanks to the ongoing development and adoption of innovative apps and digital solutions, mass volumes of data are constantly being produced and stored as business teams leverage these tools to get their work done. Leaders and their teams know that data tells the true story of their business and that it's a vital resource for driving strategic outcomes across the organization.

However, a major hurdle to overcome in the journey to widespread data and analytics adoption is the accessibility of data itself. The demand for data, analytics, and business intelligence experts is high, but if organizations want to become truly data-driven, they can no longer rely solely on talent in the data science or IT department. Every employee in every organization needs to be able to access and leverage the data they need to do their job well.

This is where conversational AI solutions that are built to enable seamless database access can offer immense value to businesses, by paving the path for instant and intuitive access to information, for everyone.

Most of us are familiar with conversational technologies in the form of chatbots and virtual assistants. These systems were developed to answer questions and address the needs expressed by users in real time, and are widely employed in digital customer service and marketing initiatives. Building on this motivation, there's a powerful business use case to be made for employing conversational AI for database access to reap the same benefits a customer might experience when they engage with a customer service or marketing chatbot: getting instant answers to pressing questions, simply by asking for them.

While current applications of conversational technologies have been successful in changing the way users are interacting with digital systems overall, we do not believe that the conversational technology that has been developed to date is powerful or flexible enough to accomplish the critical task of enabling businesses to quickly and easily get to the data that matters for decision making.

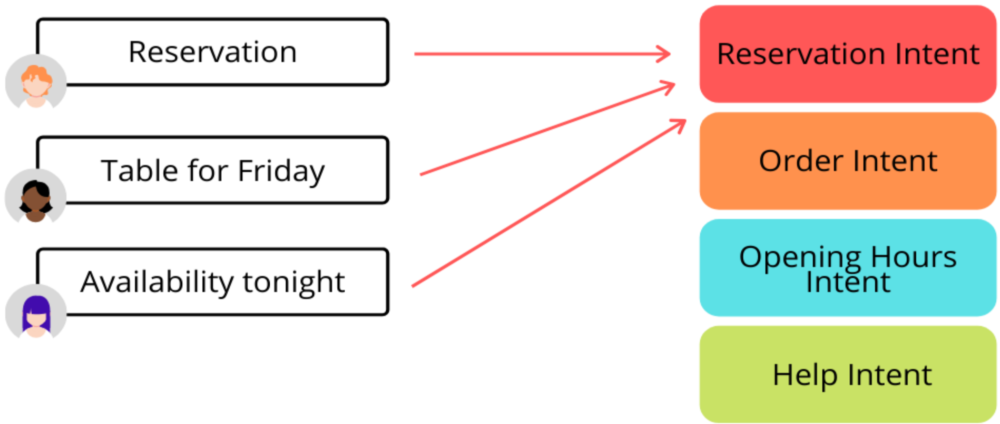
In this white paper, we'll discuss these limitations and what it takes to innovate towards a future where humans can interact with their databases the same way they interact with other humans – through conversation.

An Overview of Intent Classification

Behind the scenes of most modern chatbots is an AI intent classifier. Intent classifiers perform the function of recognizing *intent* in a user's natural language (NL) question or statement – the thing the user wants to do or accomplish – and categorizing that intent in order to offer the best possible response.

AI intent classifiers analyze statements like "I want to buy a premium subscription, is it billed monthly or annually?" These systems leverage both natural language processing (NLP) and natural language understanding (NLU) to figure out that words like *buy* or *subscription* are likely to indicate that the intent of this message is *purchase*.

The AI-driven chatbot therefore needs established intent categories in order to classify intent. These categories must be tailored to the matter at hand. If the chatbot is being used for customer service at a SaaS company, intent categories might include *needs help, demo request, downgrade, upgrade,* and *card expired*. For a hotel booking chatbot, intents would be different and include things like *book, cancel, change rooms,* and *change travel dates.*



Intent classification is most helpful when there is a limited range of intents expressed by users who typically interact with the system.

When appropriate intents have been defined for a given chatbot, the AI must be trained to correctly match or associate those intents with a variety of different words a customer might use. This is where machine learning comes in.

By processing a large volume of example data – known as training data – the intent classifier begins to learn how to match human words to the intent categories it's been programmed to know.

Intent classifiers can be used in tandem with other machine learning models that are equipped to understand some amount of context or make predictions about users' needs in order to help facilitate an efficient and rewarding user experience.

But the goal of conversational AI is to close the gap between computers and humans, not to create a mediocre substitute for human-to-human interactions.

Considering that, intent classifiers still aren't the best solution for facilitating flexible, dynamic user experiences that feel as intuitive as a conversation with another person and yield the same results: namely, getting the exact information you've asked for instantly, easily, and consistently.

Going Beyond Intent Classification

Let's start with an example of how humans typically communicate by considering the sentence "Who owes me?" This is the type of question that humans are equipped to answer, but difficult for computers to understand. "Who owes me?" is a question asked in context and, though it might refer to a specific piece of information, there's a lot of ambiguity around what that specific information might be.

In other words, the person asking the question has an idea of the information they're looking for, but they haven't clearly stated *what* is owed. Maybe they need you to tell them who has yet to pay them back for dinner last night, or maybe they're asking which of their clients is late in paying their monthly invoices. The entities that would typically be used to classify the *intent* of this statement are not present.

> Intent classifiers don't offer the flexibility, accuracy, or opportunity to access and explore data conversationally.

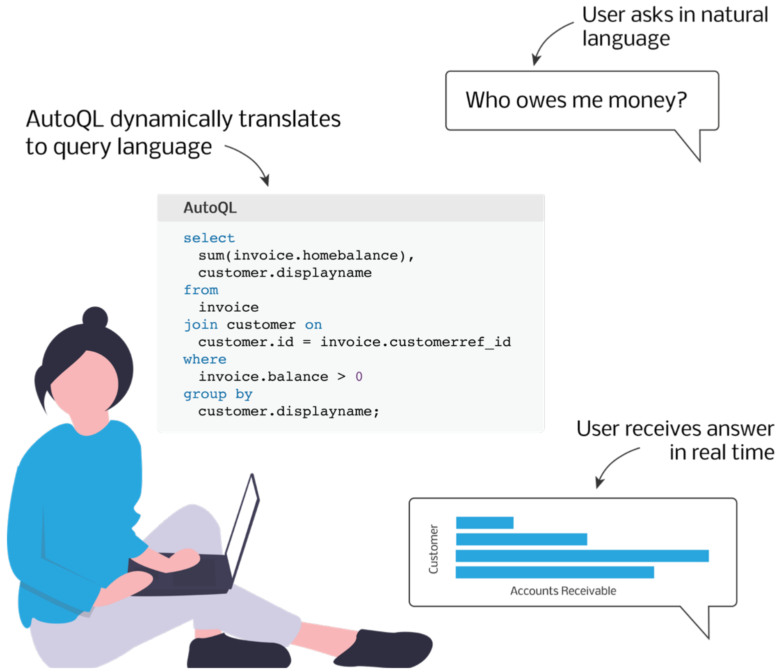
Humans can deduce intent by factoring in real-life context (what the intention might be while talking to a friend versus speaking with the accountant).

An intent classifier might have some level of context if it's built for a specific purpose, say, consumer banking, but because the intent classifier can only match a phrase to an intent that it already knows exists, this limits its flexibility. Unless every single possible intent is built into the intent classification system, there will always be gaps in the machine's understanding and ability to respond to what a human is really asking.

Another drawback of intent classifiers is limited NLU power. The AI only has to understand human language insofar as it can apply an intent category to statements that match its pre-defined list of intents. That means users still need to make sure that they ask questions in words that the AI is likely to understand.

This can sometimes feel like a tedious, even frustrating, game as users attempt to guess the words that the computer knows, while the system continuously returns the old refrain: “I’m sorry, I don’t understand what you’re asking for.”

At Chata, we see a gap in AI technology aimed at making data access faster and more intuitive, particularly when it comes to enterprise-grade databases. While intent classifiers have been built to provide better experiences in customer service and marketing channels, they don’t offer the flexibility or opportunity that is necessary for accessing data conversationally.



AutoQL goes beyond intent classification: our system understands full natural language statements and dynamically generates database query language.

This is because conversational AI built specifically for database access requires a level of complexity that can’t be achieved through the limited application of intent classifiers: it’s simply too labor and time intensive to create the massive number of intent categories database users might be interested in exploring.

Behind this limitation is the sheer volume of training data that would be needed to encapsulate the scope of an entire database, the business logic associated with that database, as well as all the types of questions users might ask about their data.

To fill this gap, it’s time to go beyond intent classification and build AI specifically for conversational *data* experiences.

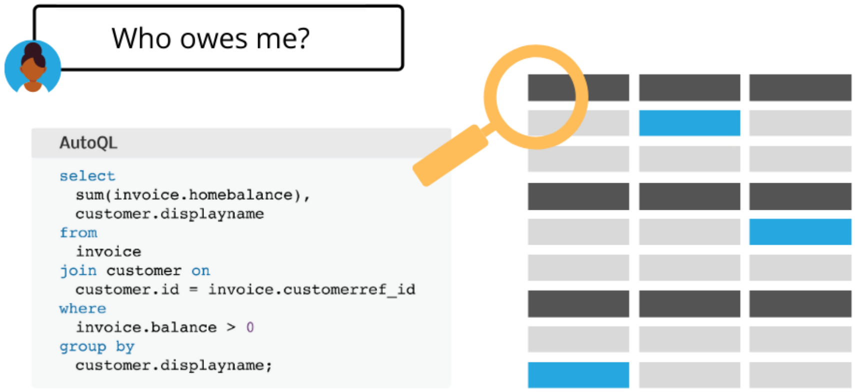
What it Takes to Build Conversational AI for Database Access

The next step forward is creating a system that can learn to understand *meaning*, not just intent, in human phrases.

Machine learning models that are trained not only to understand individual words, but that are also capable of reasoning through the relationships between those words and what they mean together, can make using a computer an even more *human* experience.

This means the computer begins to recognize that “Who owes me?” is synonymous with other questions like “Who’s in debt to me?” and that these mean the querent is asking for information about outstanding invoices (as opposed to who hasn’t e-transferred their share of the dinner bill, which might be an equally relevant question in a different context).

Teaching a machine to fully understand meaning and decipher context in natural language isn’t easy. Computers much prefer structured data, like programming language (code), to natural language, which is completely unstructured data: it doesn’t fit nicely into boxes and its rules are somewhat arbitrary.



Typically, database query language statements are used to search for specific information in a database when users are looking for answers to their questions.

It's therefore important to build AI that understands meaning in natural language, specifically in the context of database access, in order to bring the power of a dynamic and intelligent natural language-to-database query language translation system to users.

Translating Natural Language to Database Query Language

To a computer, the meaning behind natural language is largely irrelevant: it understands structured, precise commands and there's no room for nuance or ambiguity. Computers run on rules, but the rules of human language are hard for even humans to fully understand. What computers do understand is how to execute algorithms and code. They can execute actions based on the instructions given in this particular form. To get a computer system to do what you want it to do, you need to speak its language.

Because computers can't synthesize language the same way humans can, when it comes to natural language querying, an intermediary step needs to occur between a computer's understanding of the human phrase and acting on that request or query. Put simply, human language needs to be translated into the computer's language, so it can know how it should respond.

When it comes to accessing information from databases, specifically relational databases, computer systems already have "languages" for carrying out the process of searching through data and returning what the user wants to know.

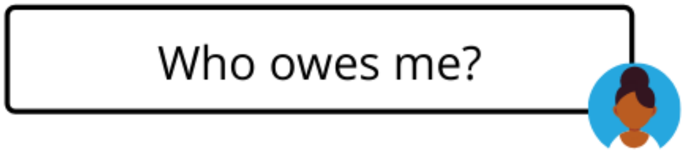
Relational databases can be searched and managed using a database query language like Structured Query Language (SQL). SQL can be written and run to identify and retrieve data from tables, columns, and rows in a database so that a user doesn't have to hypothetically scroll through thousands or even billions of data points contained in a spreadsheet or report to find out, to return to our previous example, who owes them money.

Even though database query languages exist, humans are required to learn how these languages work in order to access data. Writing database query language like SQL is a specialized skill that the average employee doesn't necessarily possess.

To bridge this skill gap and democratize the information in a database, we can employ AI to translate natural language to database query language. In other words, we can train an AI system to understand what a human is asking in their everyday language and dynamically produce a database query language statement based on that understanding.

> "Any user can simply enter a natural language query - or ask a question in their own words - to receive information from their database.

This is where the need for improved Natural Language Understanding (NLU) and advanced machine learning come in: once the machine can understand the meaning (not just the intent) of a user's phrase it should be able to dynamically create a SQL statement that reflects exactly what the user is requesting from the database.



AutoQL

```
select
  sum(invoice.homebalance),
  customer.displayname
from
  invoice
join customer on
  customer.id = invoice.customerref_id
where
  invoice.balance > 0
group by
  customer.displayname;
```

AutoQL dynamically translates natural language questions into database query language statements in real time.

With this kind of AI in place, any user can simply enter a natural language query - or ask a question in their own words - to receive information from their database.

This makes database exploration intuitive and accessible to everyone, rather than exclusive to developers or other team members who have been trained in writing database query languages. It’s a major step towards the true democratization of data, making it possible for anyone, regardless of skillset, to search any kind of aggregation of data via natural language queries and commands.

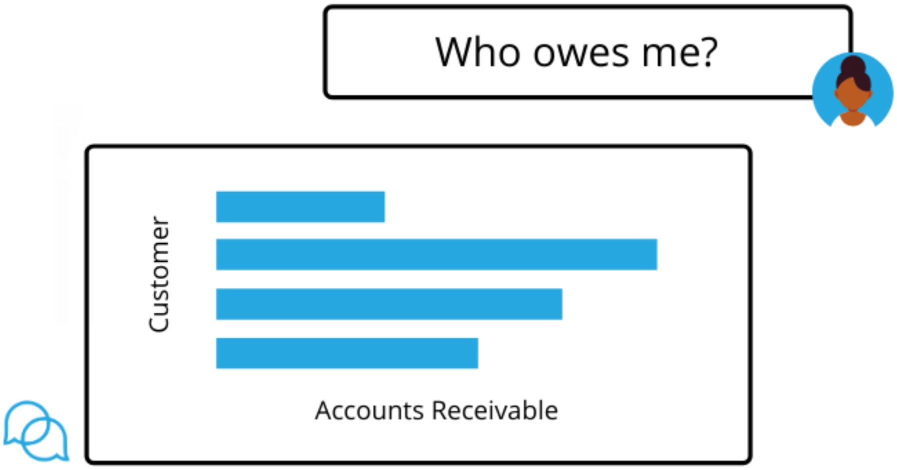
Understanding NL, Understanding SQL

To be effective, a conversational AI system designed for database access needs to deliver the outcomes that the user is expecting, every time.

While intent classification uses keywords in a user’s query to classify specific intents that can then be matched to pre-defined SQL statements, the user would have to use very specific words that the system has been trained to know.

There's also the potential for the machine to run a SQL statement that doesn't optimally reflect the user's needs, since there would only be a single SQL option associated with the user's intent. This can result in a negative user experience, one that is likely to yield mistrust of the system.

Translating natural language to database query language allows for the flexibility that humans expect from one another when asking for information. There are many ways to ask: "Who owes me?" but in the context the question is being asked, there's probably only one optimal output.



AutoQL takes into account the context of each query, understanding that the user is looking for information about their AR in this particular example.

To return an accurate and contextually correct answer, the machine needs to understand how to connect all the variables of natural language with the limited, but still extensive, variables of database query language. When a user asks: "Who owes me?" the system understands what the sequence of words mean both in whole and in part, and dynamically generates a corresponding SQL statement based on that understanding. This means multiple unique SQL statements could be generated from natural language queries that vary only slightly in how they are asked by the user.

Much like a Paris city guide fluent in English and French, the machine can interpret natural language and translate that statement to an equivalent statement in database query language to return a meaningful result.

Importantly, the way the user asked for that information might be unique or be asked in a way that the translator hasn't experienced before. But this is irrelevant because the system is designed for this purpose exactly: to decipher meaning by reasoning through the holistic natural language input to dynamically generate an optimal SQL statement that surfaces the desired result.

A system that understands both human language and database query language acts like an interpreter so that no matter how a human chooses to ask for the information they want to access in their database, the computer can dynamically generate a SQL statement that will retrieve the information that user *meant* to ask for.

Addressing the Challenges of Building Conversational AI for Database Access

There are several obstacles that must be addressed in order to successfully develop a conversational AI system for database access that's both cost-effective and powerful.

- There's a lack of available training data and the process of generating custom training data is an intensive undertaking that can consume weeks or months of the building process.

- Enterprise-grade databases are highly complex and each one holds a substantial amount of unique data.

- Human language is also inherently complex, and the system needs to be trained to understand jargon, decipher meaning, factor in context, and if necessary, verify what is being asked so users have consistently positive experiences.

Solving for the Lack of Training Data

For machine learning models to perform optimally and successfully facilitate the outcomes they were built for, they must first be exposed to enormous volumes of high-quality data. In the context of machine learning, this data is called a "training corpus" and traditionally consists of a dataset that is built manually for the specific purpose of teaching, or training, the AI model.

> The volume of training data that learning models require, though difficult and time-consuming to produce manually, can be generated through automated processes

In order for a machine to learn every way a human could ask for something *and* infer the optimal SQL statements that could be generated from those questions, the machine would need to "train on" far more natural language queries and database query language statements than are readily available.

Generating training data is a highly manual undertaking that takes an incredible amount of time and human labor to create. When it comes to enterprise-grade databases, the volume of training data required by the learning models – though difficult and time-consuming to produce manually – can be generated through automated processes. This means a high-quality customized training corpus that enables comprehensive coverage of a given database can be generated much more quickly.

In the next section, we'll talk about the profound implications of automating the process of training data generation for a faster and more robust machine learning process.

The Value of Automating Training Data Generation

Humans need to be exposed to many ideas and experiences in order to know how to behave. For a machine, this learning process is very similar, except those ideas and experiences are just data points and the relationships that exist between them. And, just like a human, the machine also needs high-quality data to learn from. If a child is continuously told that a red square is a blue circle, they won't be able to properly identify a red square in the future because they were trained on bad data. Like the child, in order for an AI system to learn correctly, it must first be exposed to good data, and lots of it. A robust and extensive training corpus that contains a high volume of accurately labeled and correctly annotated training data is absolutely vital.

> "Robust training data is absolutely essential to build a system that can truly understand and execute on users' natural language queries.

Manually generating high-quality training data takes a lot of time before the actual training of the machine models can even start. For a business to manually generate the volume and quality of training data that would be necessary to teach a conversational AI system to understand their unique database, they would need significant resources, trained specialists, and extensive machine learning expertise. By automating the process of generating training data, the machine learning models we build can get down to business much faster, saving extensively on both human labor and cost.

The more training data a machine learning model is exposed to, the better it can deliver under real-world circumstances. Therefore, having robust training data is absolutely essential in building a system that can truly understand and execute on users' natural language queries. By automating the training data generation process, the models can start training faster, leading to more immediate integration with a database and decreased time-to-value for the implementer.

Solving for Database Complexity

All databases are different: the data they hold and the structure of the database itself can vary widely from organization to organization. An AI-driven database access solution can't be one-size-fits-all because by their very nature, each unique industry (and even each individual business) necessitates unique database requirements.

Enterprise-grade databases are particularly complex and hold plenty of unique data. Because each database has distinct structure, schema, and business logic, the training corpus that is required to train the AI system for each given database must also be unique to that database. Thanks to the automated corpus generation process we've developed, it's possible to rapidly build out a robust training dataset and, from that, efficiently and optimally train a custom language model specific to a given database.

Similar to the necessary task of generating custom training data, building a custom language model for a specific database allows the AI to understand both the nuances of that database's architecture and the nuances of human language surrounding that database, inclusive of the specifics of the associated business domain.

Databases store unique data such as customer names, vendor names, and item names. A conversational AI system therefore needs to be able to understand, for example, that "Amazon" may be a customer, while in a different context in that same database, "Amazon" may also be a vendor. With great training data and a robust language model that is equipped with powerful disambiguation capabilities, the AI is able to successfully differentiate between these kinds of data points and return the appropriate data to the querent.

In the following section, we'll dive a little deeper into how this kind of system holistically handles the issue of complexity in natural language in order to continually optimize user experience.

Solving for Human Language Complexity

The custom language model not only encompasses an understanding of the structure of the database and the unique data present in said database, it's also equipped with intelligence about the ways the humans that interact with that database are likely to ask questions about it.

> To handle the nuances and complexity of human language, auxiliary machine learning models must be trained to support the conversational experience elements at play

Every industry has specific jargon, and individual businesses tend to use acronyms or terms that other businesses may not.

For example, a warehouse might use the term "work order" to describe data about a job to be completed for a client, but another warehouse might use "customer request" to describe the equivalent.

To handle the nuances and complexity of human language (and field any inconsistencies in spelling or discern meaning from terminology that the computer isn't familiar with) there also needs to be auxiliary machine learning models dedicated to supporting the conversational experience elements at play, beyond the intended capacity of the core language model.

Auxillary machine learning models work together to ensure that by the time the query reaches the translation portion of the journey, the AI fully understands what the user is asking and can return an accurate response to that query.

These models enable more seamless conversational experiences and can be implemented:

- When users are not sure what kinds of questions to ask of their data.
- When a user asks for data and the system needs to verify that it's correctly matching the words used in the NL query with the unique data the user is seeking from the database.
- To expedite the data exploration process with auto-populated suggested queries that relate to value labels specific to the database.
- To verify that what a user is asking for is what the user is actually looking for when their initial input is too ambiguous for the system to understand off the bat.

A successful conversational AI system needs to have several machine learning models trained to handle each of these circumstances as they arise.

Leveraging the training data that has been generated and working in tandem with the core language model that has been built, these auxiliary machine learning models can be trained to recognize when a user's initial input needs to be augmented in the moment to effectively generate the optimal SQL statement and return a helpful response.

AutoQL Provides the Best Conversational AI for Database Access

As a research-forward company, Chata's methods for providing the best AI solutions for database access are constantly evolving. AutoQL, our flagship solution, is built to facilitate unprecedented user experiences for people who increasingly require seamless access to data to make impactful decisions that benefit the businesses they work for.

With AutoQL, the outcome of every natural language query input is returned data that users can rely on for further reporting and analysis. We've built embeddable frontend components like the chat window-inspired Data Messenger for instant data on demand, and BI-grade Dashboards that are built using natural language queries, enabling flexible reporting and relevant overviews on the metrics that matter most.

Thanks to our cutting-edge NLP and NLU technologies, automated training data generation techniques, and proprietary machine-learning models that enable dynamic NL to SQL translation, AutoQL is designed to enable conversational dataaccessibility for even the most complex of enterprise-grade databases.

The technology we're building is our part to play in transforming the digital landscape of today and working towards a future where humans can interact with computers - and more specifically, their data - as intuitively and seamlessly as they interact with each other.